



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

File

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/040,578	12/28/2001	David J. Long	50277-1766	3641
42425	7590	06/26/2006	EXAMINER	
HICKMAN PALERMO TRUONG & BECKER/ORACLE 2055 GATEWAY PLACE SUITE 550 SAN JOSE, CA 95110-1089			PHAM, CHRYSTINE	
		ART UNIT		PAPER NUMBER
				2192

DATE MAILED: 06/26/2006

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents  
United States Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**MAILED**

JUN 26 2006

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 10/040,578

Filing Date: December 28, 2001

Appellant(s): LONG ET AL.

Attorney Stephen J. Shaw  
Reg. No. 56,442  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed March 3, 2006 appealing from the Office action mailed August 12, 2005.

**(1) Real Party in Interest**

A statement identifying by name the real party in interest is contained in the brief.

**(2) Related Appeals and Interferences**

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

**(3) Status of Claims**

The statement of the status of claims contained in the brief is correct.

**(4) Status of Amendments After Final**

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) Summary of Claimed Subject Matter**

The summary of claimed subject matter contained in the brief is correct.

**(6) Grounds of Rejection to be Reviewed on Appeal**

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

**(7) Claims Appendix**

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(8) Evidence Relied Upon**

6,385,618                    NG ET AL.                    May 7, 2002

The Java Virtual Machine Specification, The class File Format,

"<http://java.sun.com/docs/books/vmspec/html/ClassFile.doc.html>

**(9) Grounds of Rejection**

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-15 and 17-31 are rejected under 35 USC 102(e) as being anticipated by the Ng et al. reference (hereinafter "Ng").

Claims 16 and 2 are reject under 35 US 103(a) as being unpatentable over Ng in view of The Java Virtual Machine Specification reference.

**(10) Response to Argument**

As an initial matter, it should be noted that the arguments in the Appeal Brief are presented by Appellants for the first time. Accordingly, the Examiner will address the arguments, for the first time, as followed:

**Argument A1 (Brief, page 5, under section VIII):**

Appellants contend, “*Ng does not disclose associating with an instance of a class an attribute that is not in the class or any superclass of the class*” (Emphasis from original). In other words, Appellants argue that “collection Orders\_for\_Customer” of “Class Customer 420” does not anticipate associating with an instance of a class (i.e., Customer) an attribute (i.e., an instance of class Order 424) that is not in class Customer or any superclass of class Customer, since as reasoned by Appellants, “collection Orders\_for\_Customer” is in the Customer class (Brief, pages 6-8).

First, the Examiner agrees with Appellants that “collection Orders\_for\_Customer” is indeed an attribute of the Customer class. However, the Examiner respectfully submits that, “collection Orders\_for\_Customer” clearly anticipates the act of **associating** an instance of Customer class with an attribute (i.e., an instance of Order class). As pointed out by Appellants (Brief, page 10, last paragraph), in col.6:30-67, Ng et al. explicitly discloses class Customer 420 containing a “**collection of objects** representing the **orders** (i.e., instances of class Order 424) **associated** with that **particular customer**”. It is clear from this passage, that the **collection of objects** anticipates **associating** with that **particular customer** (i.e., an instance of class Customer 420) an **order** (i.e, attribute) that is not in class Customer or any superclass of class Customer since it is the **collection of objects** representing one or more **instances of class Order 424** that is defined as an attribute of class Customer 420, and not the **actual instance(s)** of class Order 424 (i.e., attributes that are not in class

Customer) that are associated with an instance of class Customer. As can be seen from FIG.4B, no attributes of Class Customer can be found, which are of type **Order**. Instead, the only **association** between each instance of Class Customer and an instance of class Order (i.e., attribute that is not in Class Customer) is the attribute Orders\_for\_Customer, which is of type **collection**. It should be noted that a variable of type **collection** in Java contains **zero** (empty) or more objects representing, in the context of Ng, **Orders** (i.e., attributes not in class Customer) and thus the **collection** containing objects is clearly defined and distinguished from **actual instances of class Order** that are being associated with each other as well as with a particular instance of Customer class by belonging to (i.e., being represented by) the same collection associated with said instance of Customer class.

**Argument A2 (Brief, page 9, under section VIII):**

Appellants contend, “Ng does not disclose associating an attribute with an instance where the association does not cause the attribute to become an attribute of the class” (Brief, page 9).

The Examiner respectfully disagrees. As discussed above in response to Appellants’ argument 1, an instance of class Order anticipates the claimed attribute that is not from the class Customer. Since, as discussed above and taught by Ng, the **collection of objects** represent the actual instances of class Order. Therefore, the association between the attribute (i.e., order) and an instance of the class Customer does not cause the actual instance of class Order to become an attribute of class

Customer. Thus, contrary to Appellants' suggesting, "the Examiner asserts that a collection is an attribute of a class, and because the collection indicates a grouping of instances of other classes, the **groupings of instances is equivalent to an attribute of another class**" (Brief, page 11, first full paragraph), the **collection**, in the context of Ng, serves to **associate** an actual instance of class Order (i.e., attribute that is not and does not become an attribute of class Customer) with an instance of class Customer.

**Argument B (Brief, page 13, first two paragraphs)**

Appellants contend, ""Ng does not teach establishing a property bundle that is associated with one or more attributes that are not in said class or any super class of said class" (Brief, page 12, last paragraph).

The Examiner respectfully disagrees. As established in the final Office Action (see page 4), FIG.4B and associated text (see col.6:32-67) of Ng explicitly teaches implementing the **foreign key** (i.e., relationship/association between instances of Customer and Order classes) using the **collection of objects** representing the orders associated with each instance of Customer class. At least in col.5:23-col.6:31, Ng explicitly teaches the object-relational mapping tool 114 allows the programmer to manipulate/customize the database tables , rows, columns, which correspond to classes, instances of the classes, and attributes of said instances respectively, and have the source code automatically updated to reflect the database changes. It is inherent that the attribute **collection of orders** associated with each specific instance of Customer class (in the source code) is inherently maintained in the database as one or

more rows (i.e., instances) of an Order class table. In other words, the **collection of orders** clearly anticipates the **property bundle** that is associated with one or more attributes (i.e., orders) that are not in class Customer or any super class of class Customer.

**Argument C (Brief, page 14)**

Appellants contend, "Ng does not teach storing within a database, objects that define said instance, said property bundle, and said one or more attributes" (Brief, page 14).

The Examiner respectfully disagrees. Again, as discussed above in response to Argument B, it is inherent in Ng's object-relational mapping system that every instance of every class (instantiated in the source code) is reflected and maintained in the database as a row in a table, and every attribute of said every instance is reflected and maintained in the database as a column associated with said row of said table. As explicitly disclosed by Ng in col.4:14-20, object-relational mapping tool integrates both automatic customizations (i.e., modification) to the source code and modifications to the database so that the programmer does not have to manually modify the source code when the database changes, or modify the database when the source code changes. Thus, it is inherent that every instance of Customer class, every collection Orders\_for\_Customer (i.e., property bundle) representing one or more orders (i.e.,

attributes) associated with said instance are defined and maintained in the database as rows, columns in tables.

**Argument D (Brief, page 15)**

Similarly, Appellants contend, “Ng does not teach maintaining an object relational mapping system that indicates a correlation between said instance and data stored in a relational database” (Brief, page 15).

Again, the Examiner respectfully disagrees. As discussed above in response to Arguments B & C, Ng clearly teaches an object-relational mapping tool between the source code and the database to integrate automatic modification of the source code to reflect database change and vice versa. Needless to say, Ng clearly teaches maintaining an object relational mapping system to indicate a correlation between every instance of Customer class and the orders associated with said instance (i.e., data stored in a relational database).

**Argument E (Brief, page 16)**

Appellants contend, “Ng does not teach establishing a pointer from said instance to a property bundle” (Brief, page 16, last four paragraphs).

The Examiner respectfully disagrees. As discussed in response to above Arguments, every instance of Customer class is associated (i.e., linked) to a collection of orders (i.e., property bundle), which are maintained in the database as one or more rows of the class Order table. Moreover, as shown in FIG.4B, class Customer defines

methods for retrieving and updating said collection of orders, namely, methods `getOrdersForCustomer()`, `addOrdersForCustomer(Order O)`, and `removeOrdersForCustomer(Order O)`. It is inherent in Ng's object-relational mapping system that each instance of Customer class has a pointer (i.e., link or reference) to the corresponding property bundle (i.e., collection of orders) associated with said instance to facilitate the updating of the database to reflect the execution of said methods.

**Argument F (Brief, page 17)**

Appellants contend, "Ng does not teach establishing a pointer from said attribute to said property bundle" (Brief, page 17, last three paragraphs).

The Examiner respectfully disagrees. As acknowledged by Appellants (Brief, page 17, second to last paragraph) and established in the final Office Action, elements Class Order 424, and Customer `Customer_for_order` of FIG.4B and associated text were cited as anticipating establishing a pointer from said attribute (i.e., an instance of class Order) to said property bundle (i.e., collection `Orders_for_Customer`). Indeed, the Examiner again, respectfully submits that, as shown in FIG.4B, every instance of Order class has an access (i.e., method to an instance of Customer class that is associated/linked with said particular Order (i.e., `Customer getCustomerForOrder()`). Furthermore, as discussed above in response to Argument E, every instance of Customer class has a pointer to the corresponding property bundle (i.e., collection of orders). Therefore, Ng clearly teaches establishing a pointer from said attribute (i.e., instance of Order class) to said property bundle (i.e., collection of orders).

**Argument G (Brief, page 18)**

With respect to claims 10 and 26, Appellants contend, "Ng does not teach maintaining a table that includes an entry that indicates that said instance is associated with said attribute" (Brief, page 18, last four paragraphs).

The Examiner respectfully disagrees. As discussed in response to above Arguments, Ng clearly discloses maintaining in the database a column (i.e., entry) corresponding to attribute collection Orders\_for\_Customer for every instance (i.e., table row) of class Customer. In turn, the collection of orders corresponds instances of class orders (i.e., attributes) which are maintained as rows in the Order table. Needless to say, the entry (i.e., column) for Orders\_for\_Customer in the Customer table, is clearly indicative of the association between each instance of Customer class and the actual orders (i.e., attributes) associated with said instance.

**Argument H (Brief, page 19)**

In response to Appellants' general argument, "by virtue of their dependence from claim 10, which is distinguished from Ng, claims 11-13 are distinguished from Ng" (Brief, page 19, third to last paragraph), in view of the foregoing discussion of claim 10 (see response to Argument G above), the rejection of claims 11-13 is deemed proper and maintained.

In response to Appellants' general argument, "claims 27-29 recite a computer-readable medium that carries instructions for performing the steps of claims 11-13,

including the features distinguished from Ng" (Brief, page 19, second to last paragraph), as established in the final Office Action (page 7), Ng discloses computer-readable medium carrying instructions for performing the steps of claims 11-13. In view of the foregoing discussion, rejection of claims 27-29 is deemed proper and maintained.

#### **(11) Related Proceeding(s) Appendix**

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

#### **(12) Text of the Final Rejection (reproduced herein for completeness)**

##### ***Claim Rejections - 35 USC § 102***

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:  
*A person shall be entitled to a patent unless –*  
*(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.*
2. Claims 1-15, 17-31 are rejected under 35 U.S.C. 102(e) as being anticipated by Ng et al. (US 6385618 B1, hereinafter Ng et al.).

##### **Claim 1**

*Ng et al. teach a method of specifying a structure (see at least 116 FIG.1 & associated text; collection Orders\_for\_Customer FIG.4B & associated text), within a computer system (see at least 100 FIG.1 & associated text), of an instance of a class (see at least table, rows, customer information, customer table col.1:60-col.2:11; int Cust\_id; 420 FIG.4B & associated text), the method including the step of:*

Art Unit: 2192

- associating with said instance of said class an attribute that is not in said class or any superclass of said class (see at least *collection Orders\_for\_Customer; int Cust\_id*; 420 FIG.4B & associated text), thereby establishing for said instance said structure that includes storage for data associated with said attribute (see at least *collection Orders\_for\_Customer; int Cust\_id*; 420 FIG.4B & associated text); and
- wherein associating said attribute with said instance does not cause said attribute to become an attribute of said class (see at least *Class 420, customer table 202, collection of objects, orders, particular customer, grouping of instances* col.6:30-67).

### **Claim 2**

The rejection of base claim 1 is incorporated. *Ng et al.* further teach wherein the step of associating further includes the steps of:

- establishing a property bundle that is associated with one or more attributes that are not in said class or any super class of said class (see at least *collection Orders\_for\_Customer; int Cust\_id*; 420 FIG.4B & associated text; *customer 302, foreign key 306, order 304* FIG.3 & associated text); and
- associating said instance with said property bundle (see at least *collection Orders\_for\_Customer; int Cust\_id*; 420 FIG.4B & associated text; *customer 302, foreign key 306, order 304* FIG.3 & associated text).

### **Claim 3**

The rejection of base claim 2 is incorporated. *Ng et al.* further teaches storing within a relational database, objects that define said instance, said property bundle, and said one or more attributes (see at least *relational database, tables, rows, customer table* col.1:60-col.2:11).

### **Claim 4**

The rejection of base claim 1 is incorporated. *Ng et al.* further teaches maintaining an object relational mapping system that indicates a correlation between said instance and data stored in a relational database (see at least *object-relational mapping tools, relational database, tables, rows, customer table, order table* col.1:60-col.2:11).

### **Claim 5**

The rejection of base claim 1 is incorporated. *Ng et al.* further teach wherein said step of associating includes establishing a pointer from said instance to a property bundle (see at least 310 FIG.3 & associated text; *iterator getOrdersForCustomer()* FIG.4B & associated text).

### **Claim 6**

The rejection of base claim 5 is incorporated. *Ng et al.* further teach wherein said step of associating includes establishing a pointer from said attribute to said property

bundle (see at least *Class Order 424, int Order\_id, Customer\_for\_Order* FIG.4B & associated text).

**Claim 7**

The rejection of base claim 1 is incorporated. *Ng et al.* further teach wherein a property class contains said attribute (see at least *Class Order 424* FIG.4B & associated text).

**Claim 8**

The rejection of base claim 1 is incorporated. *Ng et al.* further teaches associating a key with said attribute wherein said key identifies said attribute (see at least *Class Order 424, int Order\_ID, str Date* FIG.4B & associated text).

**Claim 9**

The rejection of base claim 8 is incorporated. *Ng et al.* further teaches wherein said key is a user-defined key (see at least *Class Order 424, int Order\_ID, str Date* FIG.4B & associated text; *programmer, customization* col.6:5-30).

**Claim 10**

The rejection of base claim 1 is incorporated. *Ng et al.* further teaches maintaining a table that includes an entry that indicates that said instance is associated with said attribute (see at least *Class 420, customer table 202, collection of objects, orders, particular customer, grouping of instances* col.6:30-67).

**Claim 11**

The rejection of base claim 10 is incorporated. *Ng et al.* also further teaches maintaining said entry to include a key that identifies said attribute (see at least *Class Order 424, int Order\_ID, str Date* FIG.4B & associated text).

**Claim 12**

The rejection of base claim 10 is incorporated. *Ng et al.* further teaches maintaining said table externally to said instance (see at least *Class 420, customer table 202* col.6:30-67).

**Claim 13**

The rejection of base claim 10 is incorporated. *Ng et al.* further teaches maintaining said table internally to said instance (see at least *Class 420, customer table 202, collection of objects, orders, particular customer, grouping of instances* col.6:30-67; *customer 302, hash table 310* FIG.3 & associated text).

**Claim 14**

The rejection of base claim 1 is incorporated. *Ng et al.* further teaches storing into said instance a hash table; and locating an entry in said hash table for said attribute (see at least *customer 302, hash table 310* FIG.3 & associated text).

**Claim 15**

The rejection of base claim 14 is incorporated. *Ng et al.* do not expressly disclose receiving data that is designated as a key for locating said entry in said hash table; and using said data as said key to locate said entry without determining whether said data conforms to software rules. However, this feature is deemed inherent in the teaching of *Ng et al.* since the retrieval of data stored in a hash table requires receiving a key value by which a key comparison can be made by the hash table's search algorithm.

**Claim 17**

*Ng et al.* disclose a computer-readable medium (see at least 104 FIG.1 & associated text) carrying instructions (see at least 114 FIG.1 & associated text) for specifying a structure (see at least 116 FIG.1 & associated text; *collection Orders\_for\_Customer* FIG.4B & associated text), within a computer system (see at least 100 FIG.1 & associated text), of an instance of a class (see at least *table, rows, customer information, customer table* col.1:60-col.2:11; *int Cust\_id*; 420 FIG.4B & associated text), the instructions including instructions for performing the step of:

- o associating with said instance of said class an attribute that is not in said class or any superclass of said class (see at least *collection Orders\_for\_Customer; int Cust\_id*; 420 FIG.4B & associated text), thereby establishing for said instance said structure that includes storage for data associated with said attribute (see at least *collection Orders\_for\_Customer; int Cust\_id*; 420 FIG.4B & associated text); and
- o wherein associating said attribute with said instance does not cause said attribute to become an attribute of said class (see at least *Class 420, customer table 202, collection of objects, orders, particular customer, grouping of instances* col.6:30-67).

**Claims 18-31**

Claims recite limitations, which have been addressed in claims 2-15, therefore, are rejected for the same reasons as cited in claims 2-15.

***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:  
*(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.*

4. Claims 16, and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Ng et al.* in view of The Java Virtual Machine Specification (The class File Format, <http://java.sun.com/docs/books/vmspec/html/ClassFile.doc.html>), hereinafter, *JVM Spec*.

**Claim 16**

The rejection of base claim 1 is incorporated. *Ng et al.* do not expressly disclose wherein the class is a file type and said instance is a tile of said file type in a file system wherein the step of associating includes associating with said file of said file type an attribute that is not associated with said file type or any super class of said file type. However, *JVM Spec* discloses a class is a file type and said instance is a file of said file type in a file system (e.g., see section 4.1 *ClassFile* pages 1-6) wherein the step of associating includes associating with said file of said file type an attribute that is not associated with said file type or any super class of said file type (e.g., see section *fields[]* page 5). It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *JVM Spec* into that of *Ng et al.* for the inclusion of an instance of a file type. And the motivation for doing so would have been to enable class analysis (without loading the class) and synthesis.

**Claim 32**

The rejection of base claim 17 is incorporated. Claim recites limitations, which have been addressed in claim 16, therefore, is rejected for the same reasons as cited in claim 16.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Chrystine Pham

Conferees:

Tuan Dam SPE (Art Unit 2192)

  
TUAN DAM  
SUPERVISORY PATENT EXAMINER

Wei Zhen SPE (Art Unit 2191)

  
WEI ZHEN  
SUPERVISORY PATENT EXAMINER